

BANDIT Write-up (L31 -L34)

Bandit is a character in a fictional hacking scenario on a website called OverTheWire.org.

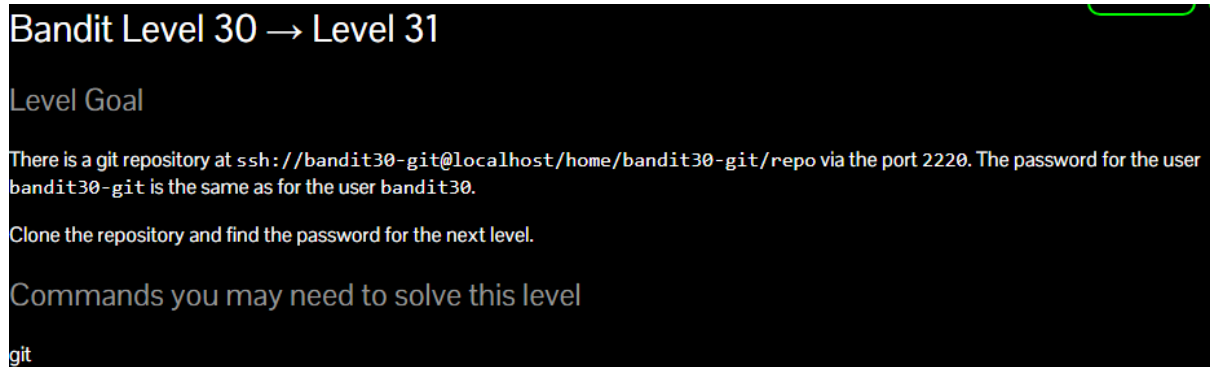
OverTheWire offers a series of wargames designed to teach cybersecurity skills in a safe environment. Bandit is the first wargame in the series, aimed at beginners. It introduces basic *Linux* commands and file manipulation through a series of challenges.

In the Bandit scenario, you play as a new user trying to gain access to higher levels by solving puzzles and cracking passwords. There is no violence or criminal activity involved.

Bandit is the suggested introductory "wargame" within the OverTheWire suite of games. It covers fundamental Linux commands and gradually progresses to advanced techniques as players advance through higher levels. Below are the walkthroughs and methodologies employed to navigate the challenges successfully.

Note for VMs: You may fail to connect to overthewire.org via SSH with a "*broken pipe error*" when the network adapter for the VM is configured to use NAT mode. Adding the setting IPQoS throughput to `/etc/ssh/ssh_config` should resolve the issue. If this does not solve your issue, the only option then is to change the adapter to Bridged mode.

Level 31:



Bandit Level 30 → Level 31

Level Goal

There is a git repository at `ssh://bandit30-git@localhost/home/bandit30-git/repo` via the port 2220. The password for the user `bandit30-git` is the same as for the user `bandit30`.

Clone the repository and find the password for the next level.

Commands you may need to solve this level

git

Clone the repo, cat the README:

```
bandit30@bandit:/tmp/anyway368/repo$ cat README.md
just an empty file... muahaha
```

This time there's nothing in the file. Git has many features, like branches and tags etc. We checked the branches, but there was nothing. Let's check the tags, which are checkpoints of version history of a repo:

```
bandit30@bandit:/tmp/anyway368/repo$ git tag
secret
```

```
bandit30@bandit:/tmp/anyway368/repo$ git show secret  
fb5S2xb7bRyFmAvQYQGEqsbhVyJqhnDy
```

fb5S2xb7bRyFmAvQYQGEqsbhVyJqhnDy

Level 32:

Bandit Level 31 → Level 32

Level Goal

There is a git repository at `ssh://bandit31-git@localhost/home/bandit31-git/repo` via the port 2220. The password for the user `bandit31-git` is the same as for the user `bandit31`.

Clone the repository and find the password for the next level.

Commands you may need to solve this level

git

```
bandit31@bandit:/tmp/anyway357/repo$ ls -la  
total 20  
drwxrwxr-x 3 bandit31 bandit31 4096 Jul 23 15:48 .  
drwxrwxr-x 3 bandit31 bandit31 4096 Jul 23 15:48 ..  
drwxrwxr-x 8 bandit31 bandit31 4096 Jul 23 15:52 .git  
-rw-rw-r-- 1 bandit31 bandit31   6 Jul 23 15:48 .gitignore  
-rw-rw-r-- 1 bandit31 bandit31 147 Jul 23 15:48 README.md
```

Clone the repo, cat the readme:

```
bandit31@bandit:/tmp/anyway357/repo$ cat README.md  
This time your task is to push a file to the remote repository.  
  
Details:  
  File name: key.txt  
  Content: 'May I come in?'  
  Branch: master
```

We need to push a file with this content, but we can't because:

```
bandit31@bandit:/tmp/anyway357/repo$ cat .gitignore  
*.txt
```

So, any file in the repo with `.txt` will be ignored, we need to delete this git ignore.

```
bandit31@bandit:/tmp/anyway357/repo$ nano key.txt
```

And add the content to it.

```
bandit31@bandit:/tmp/anyway357/repo$ git add key.txt
```

To add the file in the repo.

```
bandit31@bandit:/tmp/anyway357/repo$ git commit -m "Added key.txt"
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

To commit the changes.

Finally, just push the repo commits:

```
bandit31@bandit:/tmp/anyway357/repo$ git push
```

```
remote: ### Attempting to validate files... ###
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
remote: Well done! Here is the password for the next level:
remote: 309RfhqyAlVBEZpVb6LYStshZoqoSx5K
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
```

So, if here we only push the key.txt file with specific content inside, it will give us the flag on push.

309RfhqyAlVBEZpVb6LYStshZoqoSx5K

Level 33:

Bandit Level 32 → Level 33

Level Goal

After all this git stuff, it's time for another escape. Good luck!

Commands you may need to solve this level

sh, man

When we login bandit32:

```
--[ Tips ]--

This machine has a 64bit processor and many security-features enabled
by default, although ASLR has been switched off. The following
compiler flags might be interesting:

-m32                compile for 32bit
-fno-stack-protector  disable ProPolice
-Wl,-z,norelro       disable relro

In addition, the execstack tool can be used to flag the stack as
executable on ELF binaries.

Finally, network-access is limited for most levels by a local
firewall.

--[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

* gef (https://github.com/hugsy/gef) in /opt/gef/
* pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
* peda (https://github.com/longld/peda.git) in /opt/peda/
* gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
* pwntools (https://github.com/Gallopsled/pwntools)
* radare2 (http://www.radare.org/)

--[ More information ]--

For more information regarding individual wargames, visit
http://www.overthewire.org/wargames/

For support, questions or comments, contact us on discord or IRC.

Enjoy your stay!

WELCOME TO THE UPPERCASE SHELL
>> █
```

In this shell every text written is converted to Uppercase, this shell is called Broune Shell and we have to escape from it. After understanding this shell, I got to know that we can exit this shell by exporting /bin/bash:

```
>> $0
$ export SHELL=/bin/bash
$ echo $SHELL
/bin/bash
$ SHELL
sh: 3: SHELL: Permission denied
$ #SHELL
$ $SHELL
```

We enter a positional parameter to get the broune shell, then we'll export the value of shell in /bin/bash. And then just run Shell. That's it, we escaped the shell. Now just cat the password:

```
$ $SHELL
bandit33@bandit:~$ cat /etc/bandit_pass/bandit33
tQdtbs5D5i2vJwk08mEyYeyTL8izoeJ0
```

tQdtbs5D5i2vJwk08mEyYeyTL8izoeJ0

